

UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS

CONCEPTION ET IMPLANTATION SUR VLSI D'UN ALGORITHME DE
DÉTECTION DES PICS THERMIQUES EN TEMPS REEL

MÉMOIRE
PRÉSENTÉ
COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN INFORMATIQUE

PAR
CHARBEL BOUSTANY

SEPTEMBRE 2006

UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS

Département d'informatique et d'ingénierie

Ce mémoire intitulé :

CONCEPTION ET IMPLANTATION SUR VLSI D'UN ALGORITHME DE
DÉTECTION DES PICS THERMIQUES EN TEMPS RÉEL

présenté par
Charbel Boustany

pour l'obtention du grade de maître ès sciences (M.Sc.)

a été évalué par un jury composé des personnes suivantes:

Dr Ahmed LakhsasiDirecteur de recherche

Dr Marek ZarembaPrésident du jury

Dr Michaël Korwin-PawlowskiMembre du jury

Mémoire accepté le : 20 Septembre 2006

Remerciements

En premier lieu, je remercie Dieu pour tout; depuis le début de mes études jusqu'à la fin.

J'exprime toute ma gratitude et mes sincères remerciements à mon directeur de recherche, le professeur Ahmed LAKHSASI pour le grand soutien et l'encadrement professionnel qu'il a eu à m'apporter durant tout ce travail.

Je remercie également le professeur Marek ZAREMBA et le professeur Michaël KORWIN-PAWLOWSKI, membres du jury, pour leurs enrichissants conseils et suggestions.

Je remercie Antoine Shaneen pour ces précieux conseils techniques.

J'exprime toute ma reconnaissance à mon épouse Nisrine qui m'a aidé à surmonter moralement beaucoup des difficultés durant tous ces mois de travail.

Je dédie ce travail à ma fille, Chloé, pour son deuxième anniversaire.

Table des matières

Liste des figures	iv
Liste des tableaux	vi
Liste des abréviations, sigles et acronymes	vii
Résumé	ix
1 Introduction	1
1.1 Problématique	2
1.2 Objectifs	3
1.3 Méthodologie d'implantation.....	4
1.4 Méthodologie du test et de la vérification.....	9
1.4.1 Structure de vérification élémentaire	9
1.5 Structure du mémoire.....	10
2 État de l'art	12
2.1 Introduction.....	12
2.2 Méthode des capteurs répartis.....	13
2.3 Méthode du capteur-contrôleur.....	15
2.4 Méthode de GDS.....	18
2.4.1 Description de la méthode GDS	21
2.4.2 Détermination des équations de la position	23
2.5 Le problème inverse.....	26
2.6 Conclusion	27
3 Algorithmes de solutions des problèmes inverses	28
3.1 Introduction.....	28
3.2 Les méthodes de résolution des problèmes inverses.....	29
3.2.1 Méthode de spécification des fonctions	29
3.2.2 Méthode de régularisation.....	32
3.3 Conclusion	34

4	Conception Algorithmique	36
4.1	Introduction.....	36
4.2	Méthodologie de la conception.....	36
4.3	Stratégie de conception choisie.....	38
4.4	Types de cohésion.....	38
4.4.1	Cohésion fonctionnelle	39
4.4.2	Cohésion communicationnelle.....	39
4.4.3	Cohésion séquentielle	39
4.5	Division algorithmique	40
4.5.1	Division algorithmique fonctionnelle en classe.....	42
4.5.2	Division algorithmique temporelle	44
4.5.3	Division Fonctionnelle de l’algorithme SPTDA.....	46
4.5.4	Conclusion	50
5	Architectures internes de l’algorithme	51
5.1	Introduction.....	51
5.2	Conception modulaire de l’algorithme SPTDA.....	52
5.3	Décomposition modulaire.....	54
5.4	Architectures en pipeline	55
5.5	Architectures parallèles.....	57
5.5.1	Architectures SIMD et MIMD.....	58
5.6	Architectures Systoliques.....	60
5.7	Architecture Systolique de l’algorithme proposé	63
5.7.1	Modules 1 & 2	63
5.7.2	Modules 3.....	65
5.8	Architecture parallèle de l’algorithme.	68
5.9	Spécifications entrée/sortie.	68
5.10	Graphe de Données.....	70
5.11	Optimisation de l’architecture proposé.....	74
5.12	Conclusion	78
6	Implantation en technologie VLSI	79
6.1	Introduction.....	79
6.2	Types et approches d’implantation	80
6.2.1	Implantation dynamique et implantation statique.....	80
6.2.2	Approches d’implantation.....	81
6.3	Le FPGA	82
6.4	Programmation de modèles.....	85
6.5	Modèle de l’algorithme en Simulink	86
6.6	Simulation en virgule flottante.....	88
6.6.1	Sous-système d’entrée de l’algorithme.....	88
6.7	Résultat de simulation en virgule flottante	91
6.8	Étude de quantification	92
6.8.1	Algorithme à point fixe.....	93
6.8.2	Intervalles limites des valeurs à point fixe.....	94
6.8.3	Les erreurs de quantifications	95

6.8.4	Méthodologie de la quantification	97
6.8.5	Résultats de la simulation de la quantification	99
6.9	Conclusion	104
7	Performances et comparaisons des implantations sur FPGA	105
7.1	Introduction.....	105
7.2	La plateforme Xilinx.....	106
7.2.1	Le navigateur du projet	109
7.2.2	Résultats de l'implémentation.....	115
7.3	La plateforme Altera	117
7.3.1	Résultats de l'implémentation sur Altera.....	120
7.4	Résultats de simulation et vérification matériel - logiciel	121
7.5	Résultats de validation par rapport à la méthode FEM.....	121
7.6	Conclusion	122
8	Conclusion	124
A	Codes et rapports	127
B	Journal des erreurs et solutions	149
	Bibliographie	158

Liste des figures

Figure 1.1: Composition d'un FPGA.....	4
Figure 1.2: Structure d'un LUT.....	5
Figure 1.3: Méthodologie de conception et de simulation	6
Figure 1.4: Méthodologie utilisée (vue globale)	7
Figure 1.5: Diagramme d'un DUT.....	10
Figure 2.1: Répartition des capteurs.....	14
Figure 2.2: le détecteur de température.....	16
Figure 2.3: Débranchement temporaire d'horloge (Clock Gating)	17
Figure 2.4: Comparaison de la distribution de la température.....	18
Figure 2.5: Répartition classique.....	20
Figure 2.6: Répartition suggérée avec l'unité de contrôle.....	20
Figure 2.7: Méthode de GDS.....	21
Figure 2.8: Répartition des 2 cellules de capteurs.....	23
Figure 2.9: Schéma simplifié de la répartition des 2 cellules.....	24
Figure 3.1: Distribution des capteurs et des sources.....	32
Figure 4.1: Décomposition d'un circuit complexe en un séquenceur et un chemin de données.....	40
Figure 4.2: Division fonctionnel de l'algorithme.....	41
Figure 4.3: Diagramme de classe de la conception fonctionnelle.....	42
Figure 4.4: Fonctionnement successif du séquenceur et du chemin de données.....	45
Figure 4.5: Fonctionnement superposé du séquenceur et du chemin de données.....	46
Figure 4.6: Diagramme de séquence de la conception fonctionnel de l'algorithme SPTDA.....	47
Figure 4.7: Organigramme de l'algorithme SPTDA.....	48
Figure 4.8: Pseudo-code de l'algorithme SPTDA.....	49
Figure 5.1: Modèle architectural de l'algorithme SPTDA.....	53
Figure 5.2: Interconnexion des modules.....	55
Figure 5.3: Le pipeline exécute plusieurs instructions en parallèle.....	57
Figure 5.4: Architecture SIMD (Single Input Multiple Data).....	58
Figure 5.5: Architecture MIMD (Multiple Input Multiple Data).....	59
Figure 5.6: Réseau linéaire.....	61
Figure 5.7: Réseau orthogonal.....	61
Figure 5.8: Réseau triangulaire.....	61
Figure 5.9: Réseau hexagonal.....	61

Figure 5.10: Machines cellulaires, vectorielles et systoliques.....	62
Figure 5.11: Les processeurs systoliques.....	63
Figure 5.12: Architecture systolique de l'algorithme SPTDA proposé : pipeline en 2 branches parallèles (Modules 1 et 2).....	64
Figure 5.13: Architecture systolique du module 3.....	67
Figure 5.15: Modèle algorithmique de l'algorithme SPTDA.....	70
Figure 5.16: DFD (Data Flow Diagram) des 2 premiers modules.....	73
Figure 5.17: Architecture parallèle factorisée (1/4).....	75
Figure 5.18: Architecture parallèle factorisée (2/4).....	75
Figure 5.19: Architecture parallèle factorisée (3/4).....	76
Figure 5.20: Architecture parallèle factorisée (4/4).....	76
Figure 5.21: Performances des différentes architectures.....	77
Figure 6.1: Les FPGA entre Microprocesseurs et ASIC.....	83
Figure 6.2: Les blocs Simulink du module 1.....	86
Figure 6.3: Les blocs Simulink du module 2.....	86
Figure 6.4: Les blocs Simulink du module 3.....	87
Figure 6.5: Système de calcul (3 Modules).....	87
Figure 6.6: Le sous système d'entrée connecté avec le sous système de calcul.....	90
Figure 6.7: Comparaison entre fonction initiale et fonction calculée par l'algorithme.....	91
Figure 6.8: Division du mot en 3 parties: entière, fractionnaire et bit de signe.....	93
Figure 6.9: Exemple d'une représentation de 16 bits.....	94
Figure 6.10: Fonction de quantification trois bits (en escalier).....	96
Figure 6.11: Erreur de quantification trois bits, après centrage du signal initial.....	96
Figure 6.12: Diagramme et cycle de quantification.....	98
Figure 6.13: Algorithme de quantification.....	101
Figure 6.14: Quantification de 11 à 14 bits.....	103
Figure 6.15: Comparaison fonction initiale avec les bits de 11 à 16.....	104
Figure 7.1: Interface Xilinx niveau 1 de l'algorithme sous Simulink.....	107
Figure 7.2: Module 1 et module 2 modélisés par les blocs de Xilinx ISE.....	108
Figure 7.3: Module 3 modélisé avec les blocs de Xilinx ISE.....	109
Figure 7.4: Structure haute niveau de l'algorithme de détection.....	110
Figure 7.5: Structure du module 1 de l'algorithme de détection sous XST.....	111
Figure 7.6: Structure module 2 de l'algorithme de détection sous XST.....	111
Figure 7.7: Structure du module 3 de l'algorithme.....	112
Figure 7.8: Structure des portes logiques de l'additionneur en pipeline.....	112
Figure 7.9: Structure du "Test Bench" niveau 1.....	113
Figure 7.10: Structure du "Test Bench" niveau 2.....	114
Figure 7.11: Placement de l'algorithme sur la carte Xilinx 2v2000.....	115
Figure 7.12: Routage de l'algorithme sur FPGA Xilinx 2v2000.....	116
Figure 7.13: L'algorithme modelé avec Altera (niveau 2).....	118
Figure 7.14: Synthèse de l'algorithme (Niveau 1 - Quartus II v 4.2).....	118
Figure 7.155: Structure de portes logiques du diviseur utilisé.....	119
Figure 7.166: Structure des portes logiques du l'additionneur utilisé.....	119
Figure 7.177: Comparaison entre signal initial, logiciel et matériel.....	121

Liste des tableaux

Tableau 5-1: Opérations de base du calcul	54
Tableau 7-1: Caractéristiques matérielles du «XC2V2000 multimedia board»	110
Tableau 7-2: Pourcentage d'occupation physique.....	117
Tableau 7-3: Fréquence et délais internes de l'implantation.....	117
Tableau 7-4: Sommaire thermique de la carte utilisée.	117
Tableau 7-5: Occupation matérielle de l'algorithme sur Altera	120
Tableau 6-6: Comparaison entre températures générées par FEM , SPTDA (float) et SPTDA (fix)	122

Liste des abréviations, sigles et acronymes

ASIC	Application Specific Integrated Circuit
CLB	Configurable Logic Blocs
DFD	Data Flow Diagram
DSP	Digital Signal Processing
DTM	Dynamic Thermal Management
EDIF	Electronic Design Interchange Format
FPGA	Field Programmable Gate Array
FEM	Finite Element Method
FF	Flip Flops
GDS	Gradient Direction Sensor
HIL	Hardware In the Loop
IOB	Input Output Blocs
ISE	Integrated Simulation Environment
ITRS	The International Technology Roadmap for Semiconductors
LUT	Look Up Table
LIMA	Laboratoire d'Ingénierie des Microsystèmes Avancés
MIMD	Multiple Input Multiple Data
NGD	Native Generic Database
PE	Processus Élémentaires
SLPS	System Level Prototyping Station
SIMD	Single Input Multiple Data
SOC	System On a Chip
SPTDA	Surface Peaks Thermal Detector Algorithm

VHDL	Very Large Scale Integration Circuit Hardware Description Language
VLSI	Very Large Scale Integration
XST	Xilinx Synthesis Tool

Résumé

L'évolution de l'industrie des circuits intégrés durant la dernière décennie a été tellement rapide qu'il est possible d'intégrer actuellement des systèmes complexes sur une seule puce SoC (System on Chip). Cette évolution vers des niveaux d'intégration de plus en plus élevés est motivée par les besoins de systèmes de plus en plus performants, légers, compacts et consommant un minimum de puissance.

Cependant, l'accroissement de la densité de puissance dissipée a amené des problèmes thermiques considérables. Ainsi la chaleur produite dans la structure des semi-conducteurs est dirigée vers ses bords où elle est dissipée par le rayonnement, la conduction ou la convection. L'effet principal de l'absence d'une bonne gestion de la dynamique thermique est la dégradation graduelle et continue de la qualité de la performance ainsi que des effets directs sur la durée de vie des systèmes électroniques.

L'objectif principal du projet concerne la conception et l'implantation en technologie d'intégration à très grande échelle (VLSI – Very Large Scale Integration) d'un algorithme de détection et de localisation des pics thermiques. Ces derniers sont indispensables dans les domaines de traitement d'images thermiques et l'analyse des contraintes thermomécaniques induites au niveau des couches semi-conductrices. Les algorithmes mis en œuvre seront formulés de façon à faciliter le développement d'architectures en vue de leur implantation en VLSI. Ces architectures seront modélisées en langage de haut niveau, simulées afin d'évaluer leurs performances et implantées sur un FPGA. Les contributions scientifiques du projet

concernant l'adaptation des algorithmes de détection et de localisation afin de pouvoir les implanter sur un circuit VLSI. Également, la conception de modules de prédiction et de mesure des paramètres permettra de valider les principes de l'algorithme.

La méthode de détection utilisée est basée sur le concept de «Gradient Direction Sensor-GDS» qui est basé à son tour sur le concept du problème inverse, où on connaît les effets et on doit calculer les causes.

La validation des résultats de simulation des modèles est faite en utilisant le logiciel SimulinkTM sous Matlab[®], qui nous permet de simuler le comportement du système dans le temps. Les résultats obtenus montrent que la détection a été réalisée avec une grande précision.

L'implantation matérielle a été faite en visant 2 familles différentes de 2 producteurs différents : XilinxTM et AlteraTM, afin d'étudier la différence et d'évaluer la performance sur chacune des cartes.

Abstract

The rapid evolution of the industry of the integrated circuits during the last decade was so quick that currently it is possible to integrate complex systems on a single SoC (System on a Chip). This evolution towards higher integration levels is motivated by the needs of advanced high performance, lighter and more compact systems with less power consumption.

Nevertheless, the growth of power density dissipated brought a number of critical thermal problems. Thus the heat produced in the structure of the semiconductors is directed towards its edges where it is dissipated by radiance or convection. The principal effect of the absence of a good dynamic thermal management is the gradual and continuous degradation of the quality of performance as well as some other direct effects on the life cycle of the electronic systems.

The main objective of the project concerns the design and the implementation in (VLSI – **Very Large Scale Integration**) of an algorithm for the detection and the localisation of the thermal peaks. These peaks are found in the domain of thermal image processing and the analysis of the induced thermo-mechanical stress. The main algorithm will be formulated in a manner to facilitate the development of architectures in regards of their implementation in VLSI. These architectures will be modeled in languages of high level, simulated in order to evaluate their performances and then implemented on a FPGA.

The main scientific contributions of the project, concerns the adaptation and optimisation of detection algorithms in order to be able to implement them on a VLSI chip as well as the proposition of an optimal parallel architecture for the algorithm developed.

The hardware implementation is formulated and adapted to different FPGA providers: XilinxTM and AlteraTM in order to study the performance of the algorithm on the 2 different hardware platforms and to evaluate the impact of the difference between these 2 implementations.

Equally, the design of special prediction units and test benches will allow us to integrally validate the principles of the algorithm.

Different new methodologies of design and implementation have been presented and compared. These methodologies are the newest in the market of VLSI design and our study shows clearly their performance.

Chapitre 1

Introduction

Les améliorations continues dans la technologie des micro-processeurs ont donné aux concepteurs des circuits intégrés la liberté et la faisabilité pour augmenter la densité de logique implantée. Cependant, cette augmentation dans la densité de logique, combinée avec la fréquence augmentée des opérations de calcul complexe a eu pour résultat une augmentation exponentielle dans la densité du courant. Selon ITRS (*The International Technology Roadmap for Semiconductors*) la densité du courant des microprocesseurs est estimée d'atteindre 100 W/cm^2 au delà de la node technologique de 50 nm [16]. Cette augmentation exponentielle de la densité du courant se manifeste sous forme de chaleur qui doit être continuellement contrôlée afin d'assurer l'opération fiable souhaitée.

Un autre facteur compliquant le problème est la présence de points chauds locaux, qui se présentent en raison des niveaux variables de dissipation du courant à travers la puce. La température accrue des jonctions et la présence de points chauds locaux sont très nuisibles au fonctionnement des puces électroniques et ont comme effet directe la diminution de la vitesse de commutation des transistors, réduisant de cette façon la performance globale du circuit.

Dans les dernières années, en raison de la densité croissante du courant dissipé dans les circuits intégrés, les problèmes thermiques sont devenus de plus en plus urgents dans le

micro-électronique moderne. La chaleur produite dans la structure des semi-conducteurs est dirigée vers ses bords où elle est dissipée par le rayonnement, la conduction ou la convection.

L'effet principal de l'absence d'une bonne gestion de contrôle thermique est la dégradation graduelle et continue de la qualité de la performance ainsi que des effets directs sur la durée de vie des systèmes électroniques.

1.1 Problématique

Les techniques sophistiquées de mise en boîtier «Packaging» sont maintenant employées pour réduire les effets de l'augmentation de la température. Une gamme de techniques allant des dispositifs d'étalement (*Heat spreaders*) et radiateurs avec convection forcée de la chaleur (*Heat sinks*) au refroidissement actif sont employées pour étaler la chaleur d'une manière plus efficace et plus fiable. Cependant ces techniques ont des effets négatifs limitant leur performance globale dus à leur forme qui est basée sur des conduits programmés du flux thermique induisant des délais et une augmentation de la température non prévues par les programmeurs des langages de haut niveau.

Dans la nouvelle génération des microprocesseurs ultra-performants, les coûts du «packaging» s'élèvent de \$1 à \$3 par chaque Watt de puissance dispersé [16]. Pour cette raison et pour maintenir la performance dans la pire des conditions d'opérations, les nouveaux microprocesseurs emploient certains façons dynamiques de gestion thermique (*Dynamic Thermal Managment- DTM*) comme par exemple la technique de consigne d'horloge (*Clock Gating*) et l'échelonnement dynamique de la tension (*Dynamic Voltage Scaling*). Le concept de DTM est automatiquement engagé lorsque la température sous contrôle dépasse une certaine valeur prés-définie.

Les «Systèmes-sur-seule-puce» (*System on a Chip- SOC*) basés sur les FPGA (Field Programmable Gate Array) sont devenus de plus en plus populaire dans l'industrie de la micro-électronique. Cette évolution exponentielle a générée en parallèle beaucoup des

problèmes en ce qui concerne le contrôle et la gestion des courants de fuite au niveau des couches semi-conducteurs à grande densité. Les SOC sont employés sous une grande variété de conditions d'opération souvent non prévus, et par conséquent, une solution qui se relie sur une technique de packaging unique sera nécessairement inefficace et inutile, d'où la nécessité critique de l'implémentation de quelques formes de DTM pour assurer l'efficacité du coût et la fiabilité d'opération.

Dans notre mémoire, nous présentons la conception et l'implantation en technologie ITGE (VLSI – *Very Large Scale Integration*) d'un système qui détecte et localise les pics thermiques dans le but de contrôler continuellement les niveaux de température aux divers emplacements sur la puce, et activer une réponse appropriée quand une urgence thermique est détectée. L'algorithme de détection sera basé sur le concept des problèmes inverses ainsi que sur la méthode des «détecteurs de la pente de la direction» (GDS- *Gradient Direction Sensor*) et optimisé en vue de son implémentation sur un FPGA (*Field Programmable Gate Array*).

1.2 Objectifs

Le but du projet est de concevoir et implémenter en technologie d'intégration à très grande échelle (VLSI) un algorithme de détection et de localisation des pics thermiques. Ces derniers jouent un rôle important dans les domaines de traitement d'images thermiques et l'analyse des contraintes thermomécaniques induites au niveau des couches semi-conducteurs. L'algorithme sera conçu en se basant sur les concepts de la résolution des problèmes inverses et sur la méthode du GDS pour faire une bonne estimation des coordonnées géométriques des pics thermiques et de leurs valeurs correspondantes. L'algorithme sera mis au point pour le problème général de détection de plusieurs sources. Les algorithmes mis en œuvre seront formulés de façon à faciliter le développement d'architectures en vue de leur implantation en VLSI.

La proposition d'une architecture parallèle nous permettra d'avoir un très bon débit afin de répondre aux besoins des calculs excessifs de la gestion de la dynamique thermique en temps-

réel tout en diminuant les opérations des consignes d'horloge (*Clock Gating*) et l'échelonnement dynamique de la tension (*Dynamic Voltage Scaling*) généralement utilisés dans des cas similaires pour optimiser la performance générale ainsi que la durée de vie du système. Cette architecture sera modélisée en langage de haut niveau, et simulée afin d'évaluer son performance et finalement implantée sur un FPGA.

Les contributions scientifiques du projet concernent l'adaptation des algorithmes de détection et de localisation afin de pouvoir les implanter sur un circuit VLSI. Également, la conception de modules de prédiction et de mesure des paramètres permettra de valider les principes de l'algorithme. Fiabilité, précision et rapidité sont les principaux objectifs qui ont guidés la réalisation du projet.

1.3 Méthodologie d'implantation

Dans notre travail, nous avons choisi d'implanter notre algorithme sur un FPGA.

Un FPGA consiste en une matrice à deux dimensions de blocs logiques reconfigurables (CLB-*Configurable logic blocs*) entourée par des blocs d'entrées/sorties (IOB-*Input output blocs*) et une matrice de connecteurs qui connecte les CLB (figure 1.1).

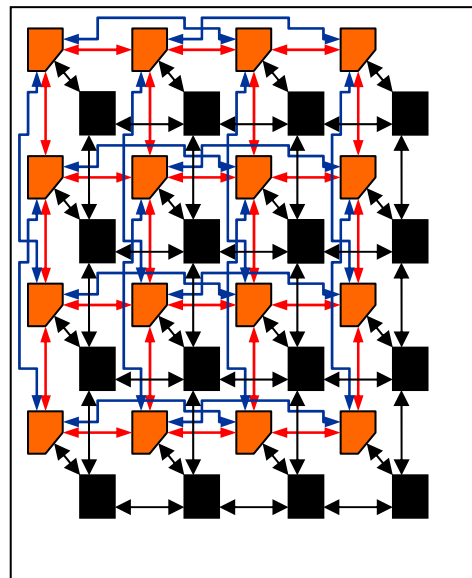


Figure 1.1: Composition d'un FPGA

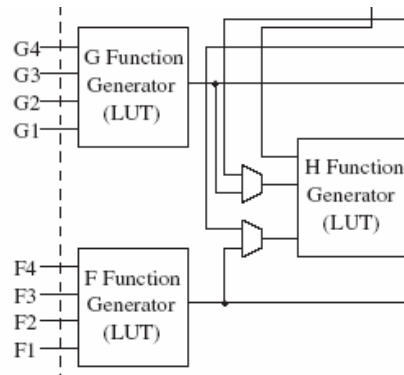


Figure 1.2: Structure d'un LUT.

Un FPGA peut être programmé en téléchargeant un fichier de bits (*bitstream*) créé par les outils de synthèse, de placement et de routage. Notre conception commencera avec la description fonctionnelle (spécifications exécutables) des fonctions de l'algorithme SPTDA (*Surface Peaks Thermal Detector Algorithm*), ensuite une programmation de modèle représente les différents modules et itérations de l'algorithme. La 3^{ème} étape c'est de manipuler nos modèles avec les outils de conception relatifs au fournisseur spécifique de la famille de FPGA à utiliser. Les outils de Synthèse et d'implémentation fournis par le constructeur du chip FPGA génèrent le code VHDL du modèle et le fichier de bits de configuration. Toutefois une série de simulation doit être exécutée afin de valider le codage et le design avant de passer d'une étape à une autre et entre toutes les étapes surtout après les modifications manuelles des codes générés par ces outils spécifiques. Une comparaison entre la simulation logicielle et la simulation matérielle en utilisant la technique du HIL (*Hardware In the Loop*) nous permet de faire une bonne comparaison de performances entre les 2 plateformes différentes. Le procédé tout entier peut être modélisé par les 2 diagrammes représentés par les figures 1.3 et 1.4.

D'autre part, toute méthodologie de conception utilisée doit prendre en considération les différentes contraintes telles que le coût, la consommation et le temps de mise sur le marché. Par exemple les développeurs des applications de traitement du signal ont recours à une mise en œuvre des nouvelles méthodologies d'implantation très efficaces.

Dans notre travail, la nature de l'algorithme étudié nécessite une méthodologie de synthèse d'architecture sous contrainte de précision et de vitesse. Donc, pour obtenir une implantation efficace, le processus d'optimisation est couplé avec la synthèse d'architecture. De plus, l'évaluation de la précision est réalisée par une approche analytique permettant ainsi d'obtenir des temps d'optimisation raisonnables.

onc, il est nécessaire d'étudier l'influence de la période d'échantillonnage et de la limitation imposée par le calcul à virgule fixe. Pour cela, une procédure de simulation est requise pour bien sélectionner la meilleure combinaison de nombres de bits dans l'étude de quantification.

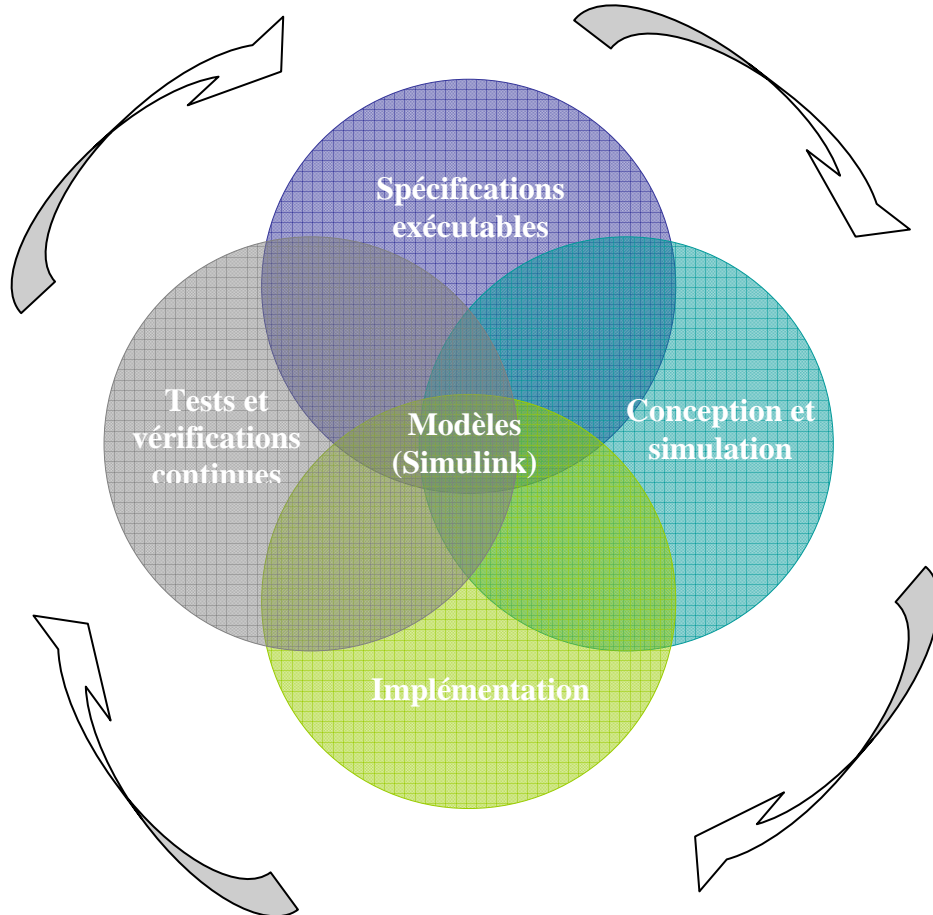


Figure 1.3: Méthodologie de conception et de simulation

La méthodologie consiste à analyser et préciser les spécifications exécutable de l'algorithme à implanter. La programmation de modèles constitue la 2^{ème} étape. Une série de simulation est nécessaire pour optimiser l'architecture modulaire utilisée. Les tests et vérifications constituent des étapes intermédiaires continues reliant tous les étapes ensemble. La phase finale d'implantation sera réalisée après la simulation algorithmique et une étude de quantification.

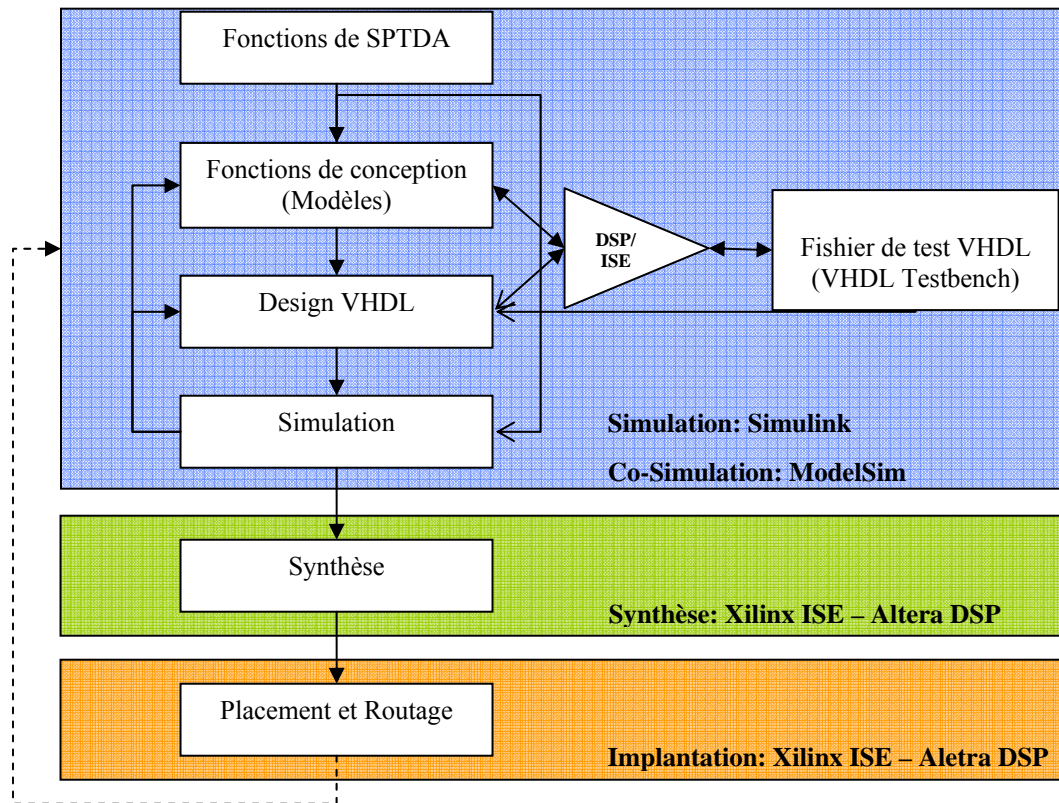


Figure 1.4: Méthodologie utilisée (vue globale)

Notre flot de conception sera divisé en 3 parties principales: simulation, synthèse et implémentation. Nous allons présenter une description détaillée de chaque partie.

- **Simulation:** nous allons tout d'abord établir des diagrammes d'états représentant les différentes fonctions de l'algorithme de SPTDA et ceci sera réalisé en utilisant la programmation de modèles, qui est très importante pour bien structurer notre algorithme en le divisant en différents modules ayant une structure hiérarchique,

suivant la technique de «diviser pour régner» (*Divide and conquer*) [3] et le tout sera fait en utilisant une architecture «systolique» [10]. Ensuite les outils XilinxTM ISE[®] (*Integrated Simulation Environment*) et AlteraTM DSP[®] (*Digital Signal Processing*) builder, intégrés dans MatlabTM Simulink[®] génère le fichier du code (VHD), par là on établit une co-simulation de ce code pour étudier son comportement fonctionnel, les modifications manuelles seront performées dans cette étape avant la simulation logicielle finale. Par la suite on établit les bancs d'essais «Test Benches» et on engage le «Force File» ainsi préparé pour exciter les paramètres intérieurs et extérieurs de l'algorithme. Ainsi le «Test Bench» testera toutes les fonctions clés du design.

- **Synthèse:** c'est l'étape qui transforme la description HDL d'une puce en portes logiques en utilisant les contraintes spécifiées par l'utilisateur (temps, variation d'horloge, surface, etc..) pour implémenter et optimiser la conception proposée dans les blocs fonctionnels primitifs sur le FPGA (les flip-flops, les portes logiques, etc.). De la synthèse, nous pouvons obtenir aussi, par exemple, la fréquence de base, et les informations techniques de la puce (secteurs et connecteurs utilisés). Cependant, la synthèse donne seulement des valeurs estimées [8]. Afin d'obtenir des résultats plus précis, l'étape de placement et routage doit être faite dans la phase d'implémentation en ciblant un FPGA bien spécifié.
- **Implémentation :** Dans notre travail, nous avons choisis d'établir une comparaison d'implantation entre 2 grandes fournisseurs de FPGA : XilinxTM et AlteraTM à cause de leurs très grandes capacités en blocs logiques, leurs grandes capacités de reconfigurabilité ainsi que leur très hautes vitesses. Les outils de synthèse, de placement et routage de Xilinx et d'Altera seront utilisés. Après la synthèse, l'outil ISE à titre d'exemple exporte un fichier EDIF (*Electronic Design Interchange Format*) qui représente le caractère fonctionnel de la conception. Le logiciel Xilinx ISE prendra ce fichier et traduira alors les listes de fonctions à l'intérieur en un fichier NGD (*Native Generic Database*) qui contient la description logique de la conception qui cible une famille de FPGA de Xilinx bien déterminée. Le design est transformé

maintenant en une série de CLB, FF et d'IOB. Après cela, la conception peut être placée sur la puce et peut être connectée (routage). Des données plus précises à propos de l'utilisation et de la performance seront fournies par le rapport final généré automatiquement par l'outil après le placement et routage.

1.4 Méthodologie du test et de la vérification

La vérification du système est faite pour valider le code de description en fonction des spécifications, c'est le processus qui tente de prouver que le modèle HDL correspond aux spécifications. Elle est faite à plusieurs moments (entre autres):

- avant-synthèse (vérification fonctionnelle).
- après-synthèse (vérification temporelle).
- après placement-routage (vérification physique).

Étant donné la complexité des circuits modernes, le processus de vérification devient de plus en plus complexe. Diverses méthodes ont été proposées afin de faciliter le travail de vérification. Dans notre projet, nous nous limitons aux méthodes de vérification élémentaire, surtout utile pour le débogage.

1.4.1 Structure de vérification élémentaire

Typiquement, la structure de test comprend trois modules:

1. Un générateur de test qui envoie des stimuli aux modules sous test.
2. Le module sous test qui est généralement désigné par DUT (*Device Under Test*). C'est un modèle décrit à l'aide d'un HDL et qui réagit aux stimuli.
3. Vérificateur de test: vérification de la réponse du module aux *stimuli*.

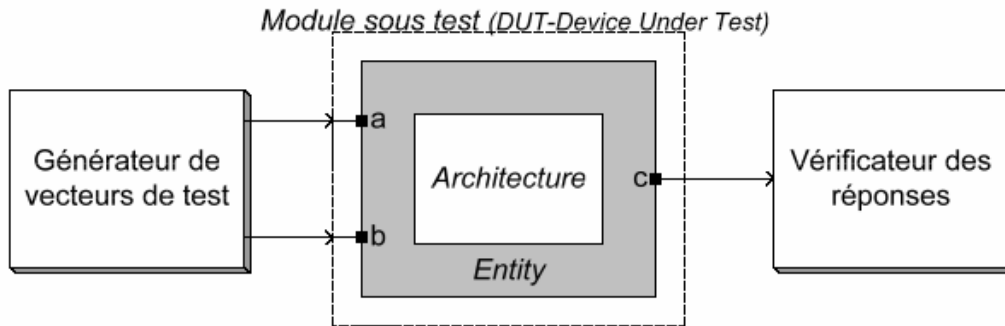


Figure 1.5: Diagramme d'un DUT

1.5 Structure du mémoire

Le deuxième chapitre présentera une synthèse des connaissances acquises par la revue de littérature dans le domaine de la détection et de la gestion des pics thermiques. Étant donné que notre algorithme est basé sur la méthode GDS et cette dernière est basée elle-même sur la technique des problèmes inverses. Nous présenterons une explication bien détaillée des avantages et des inconvénients des problèmes inverses et des algorithmes qui les traitent.

Le troisième chapitre contiendra une étude sur les algorithmes de solution des problèmes inverses ; leurs structures, principes, avantages, inconvénients. En plus, nous introduisons l'extension algorithmique à ces techniques qui traite plusieurs sources de chaleur en même temps

Le quatrième chapitre présentera le développement de la conception algorithmique. En effet, la méthodologie et la stratégie de conception utilisée, les différents types de cohésions présentées dans notre algorithme. Les aspects fonctionnels et temporels de l'algorithme proposé.

Dans le cinquième chapitre, nous présenterons les architectures modulaires sous différentes formes (parallèles, en pipeline, systolique et leurs implantations). On développera les graphes de données de l'algorithme, et on terminera par la proposition de quatre architectures différentes pour notre algorithme en vue de son optimisation.

Le sixième chapitre portera sur la programmation de modèles de l’algorithme proposé et les approches d’implantation utilisées. Nous présenterons une étude de quantification pour la détermination des nombres de bits optimaux pour l’implémentation.

Le septième chapitre portera sur les résultats de simulation des modèles VHDL, et en visant deux plateformes différentes (Xilinx™ et Altera™). Les différentes étapes de l’implantation seront présentées avec les résultats des simulations sur le matériel comparés avec les simulations logiciels.

Finalement, nous présenterons une discussion des principaux résultats, de la contribution du projet et des travaux futurs.

Chapitre 2

État de l'art

2.1 Introduction

En considérant que les effets quantitatifs de chaleur sur les systèmes électroniques ont besoin d'une soigneuse révision, il est accepté que les hautes températures ainsi que les variations thermiques réduisent la vie et la fiabilité de couches semi-conductrices.

Dans l'ère des FPGA, les technologies actuelles limitent la température maximum à 125 degrés centigrades. Pour une véritable application, la vérification de cette restriction n'est pas une tâche facile. Le concepteur doit tenir compte de beaucoup des aspects comme les tensions maximums, l'effet de dissipateurs thermiques et les ventilateurs, la position de la plaque, la température de la chambre, ou l'influence d'appareils chauds situés près du fragment étudiée.

Certes, il y a eu plusieurs travaux de recherche sur la détection et la gestion des pics thermiques [2] [6] [8] [16] [18]. Ces recherches se ressemblent par le but général mais se diffèrent extrêmement par les techniques utilisées et la méthodologie appliquée et le tout dépend des conditions considérées dans l'étude. Ces conditions ont été présentées dans le

paragraphe précédent et chacune des ces conditions affecte directement d'une manière ou d'une autre les résultats obtenus.

Une bonne étude a été faite dans [8] et une nouvelle technique de DTM a été introduite. Le travail consiste à placer un tableau de 4 par 8 détecteurs dans le FPGA. Il faut bien noter ici que les détecteurs ou capteurs utilisés sont des unités logiques programmées sur le FPGA, c.a.d. ce ne sont pas des capteurs électriques attachés sur la carte que se soit des capteurs analogiques ou numériques.

2.2 Méthode des capteurs répartis

Buedo-Boemo [8] ont présentés un nouvel concept pour concevoir ces détecteurs.

En assurant que le détecteur implanté en JBitsTM (*Xilinx Inc*) est compatible avec la reconfiguration dynamique au temps d'exécution qui est une propriété fondamentale du FPGA. Le tableau des détecteurs peut être construit par une boucle qui crée des instances du premier détecteur autant de fois que nécessaire.

Le code source Java de la boucle est présenté comme suit:

```
for (i=0; i<numOfSensors; i++)
{
  sensor[i] = new TemperatureSensor("Sensor", clk);
}
for (i=0; i<numOfSensors; i++)
{
  sensorOffset[i] = sensor[i].getRelativeOffset();
  sensorOffset[i].setHorOffset(Gran.CLB, sensorXPos[i]);
  sensorOffset[i].setVerOffset(Gran.CLB, sensorYPos[i]);
}
for (i=0; i<numOfSensors; i++)
{
  sensor[i].implement(TimeEnable, RingEnable, CountEnable);
}
Bitstream.connect(clk);
```

La technique consiste à analyser les données de ces détecteurs qui sont pré-calibrés d'avance, et par retro-configuration on contrôle les parties de l'application qui génèrent le plus d'énergie thermique. Les données sont dirigées dans des blocks RAM ou elles sont analysées après par un logiciel spécialisé. Le concepteur obtiendra un graphe thermique

reliant la limite thermoélectrique aux différents modules de l'application. Donc, le cœur de cette étude est la propriété de reconfiguration au temps d'exécution spécifique aux FPGA.

Afin de vérifier la technique, un grand circuit réel et complexe a été utilisé comme un cas d'étude: deux microprocesseurs PLASMA de 32 bits en interaction ont été étudiés par cette technique en utilisant une matrice de 32 capteurs uniformément repartis sur le chip comme nous montre la figure 3.1. Les microprocesseurs sont placés sur un FPGA de la famille Virtex de Xilinx modèle XCV800HQ240-4C.

En utilisant l'outil de placement de Xilinx, la position du premier processeur est limitée au premier 32 colonnes gauches et le second occupe les autres 32 colonnes droites.

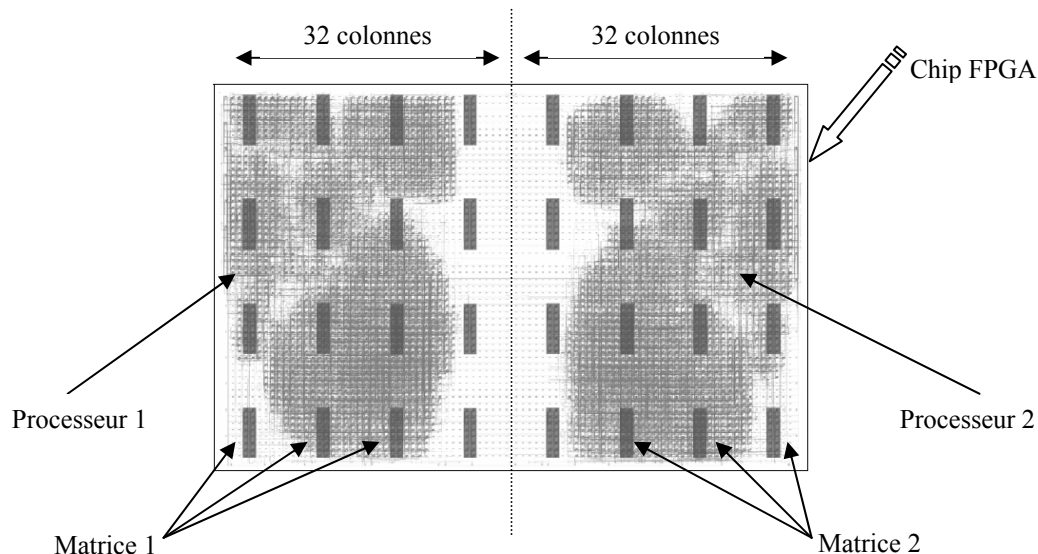


Figure 2.1: Répartition des capteurs

(Les 2 matrices superposées des détecteurs et les deux processeurs jumeaux (les obscurités foncés) FPGA Virtex XCV800HQ240-4C).

Les mesures ont été faites dans une pièce propre et isolée [8] qui garantit que les variations de température ambiantes sont toujours plus petites que 1° Celsius. De plus, l'analyse des résultats a montrée que la technique est bien sensible même aux faibles consommations d'énergie impliquées dans les expériences.

Cependant, la description des processeurs utilisés dans les expériences n'a pas d'indication de placement et par conséquent le statut thermique des différentes unités fonctionnelles ne peut pas être déterminé. Cette information serait utile pour concevoir les blocs qui produisent des points chauds.

Donc l'avantage principal consiste dans l'analyse des activités des différents blocs d'un circuit complexe implanté sur FPGA. Ceci pourrait être utile pour le concepteur des circuits électroniques qui pourrait savoir quel bloc consomme le plus de puissance, et ainsi doit être reconçu, au lieu d'appliquer une stratégie globale pour réduire la contrainte (par exemple, abaissement de la fréquence d'horloge).

Le désavantage principal est que la technique utilise des blocs logiques du FPGA elle-même. Donc pour un nombre de détecteurs plus grand, la technique seule occupe une partie considérable des ressources disponibles sur le FPGA. L'extension de cette technique sur plusieurs FPGA interconnectées n'est pas évidente. Cette technique ne peut en aucun cas, localiser géométriquement la source de chaleur la plus active sur la plaque.

2.3 Méthode du capteur-contrôleur

La deuxième étude importante a été faite dans [16]. L'étude consiste en une conception d'un système qui contrôle les températures aux divers emplacements sur le FPGA. Ce système est composé d'un contrôleur lié à un réseau de capteurs de température configuré en tableau et qui sont appliqués sur le FPGA. Les auteurs ont étudiés les problèmes impliqués dans la conception et interfaçage des détecteurs. Les valeurs fournies par les détecteurs sont comparées avec les valeurs obtenues d'un logiciel de simulation thermique nommé (*HotSpot*), qui est un outil d'analyse thermique au niveau architectural.

Le design a été implanté sur un chip FPGA de la famille Virtex-2 Pro de Xilinx (XC2VP7), mais avec des modifications mineures la technique peut être appliquée sur d'autres catégories de FPGA.

La technique consiste en une matrice de capteurs implémentée sur le FPGA et un contrôleur qui lit les différentes données des capteurs et engage le mécanisme de DTM approprié. Nous allons décrire la conception des détecteurs et des contrôleurs utilisés et faire une analyse critique de cette étude, ainsi qu'une comparaison avec la précédente et enfin introduire notre technique.

Le concept du détecteur de température utilisé est basé sur la même proposition introduite par Buedo-Boemo [8]. Ceci utilise le fait que la vitesse de commutation d'un transistor est directement proportionnelle à la température. Dans un tel détecteur, un oscillateur en anneau est implémenté sur la puce FPGA et la fréquence d'oscillation est utilisée comme un indicateur de la température.

Un oscillateur en anneau (*Ring Oscillator*) est une séquence impaire d'inverseurs connectés en série pour former une chaîne. Ceci peut être implémenté sur le FPGA en configurant les LUT (*Look Up Tables*) comme des inverseurs et les enchaînant ensemble. L'architecture d'un LUT est montrée dans la figure 4.2. Alors que le schématique du détecteur utilisé est montré à la figure 3.2.

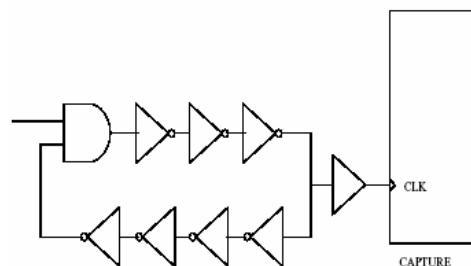


Figure 2.2: le détecteur de température

(La fréquence d'oscillation est directement proportionnelle à la température de jonction)

L'unité de contrôle détermine quand les détecteurs doivent être activés, exécute les étapes nécessaires correspondants et lit les valeurs fournies par ces détecteurs. Une fois le contrôleur saisit les températures aux divers emplacements sur la puce, il peut décider quelle technique de gestion thermique DTM doit être appliquée le cas échéant.

Dans cette étude la technique DTM utilisée est le débranchement temporaire d'horloge (*Clock Gating*), elle est basée sur le principe suivant:

Quand la température de la puce au niveau de la jonction augmente au delà d'une température limite T_i appelé «le point de déclenchement», l'horloge est retardée pour un intervalle de temps spécifié. Après que la couche se refroidit, l'horloge est réactivée, et le système continue de fonctionner. Ainsi ils échangent une partie de la performance pour une répartition plus sécuritaire de la température. Au niveau matériel, l'horloge est attachée à la porte logique et sera branchée ou débranchée suivant les bits arrivant aux entrées de cette porte. Par exemple dans la figure 3.3, A et B sont des signaux de contrôle, si $A = 1$ et $B = 1$ alors l'horloge passe à Z, si non l'horloge est bloquée.

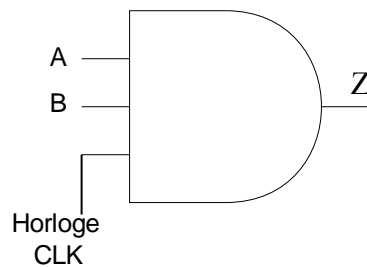


Figure 2.3: Débranchement temporaire d'horloge (Clock Gating)

Les 2 schémas ci-dessous montrent une comparaison de la distribution de l'activité de la température sur un FPGA avec et sans la technique proposée.

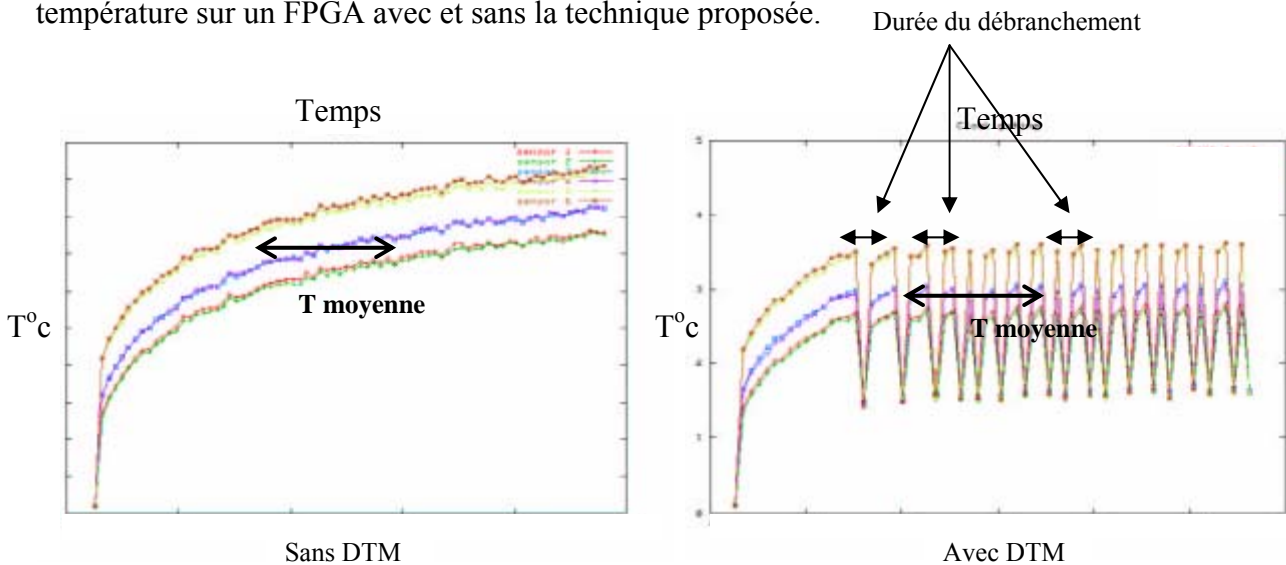


Figure 2.4: Comparaison de la distribution de la température

Sans et avec DTM respectivement. La température moyenne avec DTM est inférieure que celle sans DTM

Cette technique implémente bien le concept de débranchement d'horloge. On peut voir clairement que la moyenne des températures observées avec l'utilisation du DTM est bien inférieure à la moyenne obtenue sans DTM. Donc, dans cette étude la conception des capteurs logiques a été implantée sur le FPGA. Un contrôleur traduit les variations des fréquences des oscillations en des variations de la température et ensuite engage le protocole DTM approprié. Cette technique est efficace dans le cas des circuits moins complexes et moins chargés, car l'utilisation fréquente du débranchement d'horloge réduit considérablement la performance. Il n'y a pas un nombre optimum de détecteurs à utilisés mais si le nombre augmente, d'importantes ressources sur le FPGA seront occupées et cela peut causer des sérieux problèmes aux concepteurs.

2.4 Méthode de GDS

L'approche utilisée dans notre travail, se base sur le concept de la solution des problèmes inverses déjà introduit. Nous allons analyser et discuter les différents travaux déjà faits sur ce

sujet et nous allons discuter l'algorithme que nous proposons, ensuite on va explorer la méthodologie d'optimisation, de développement, et de codage qui va être utilisée afin d'implémenter cet algorithme sur un FPGA approprié. Nous allons présenter notre choix du FPGA qui sera utilisé et les raisons qui le soutiennent.

Jusqu'à maintenant nous avons vu que les études reliées à notre sujet sont basées sur l'implantation logique des capteurs sur le FPGA elle-même. Dans notre travail nous allons utiliser des capteurs thermoélectriques séparés de la carte, et l'algorithme gérant ces capteurs sera implanté sur un circuit VLSI (FPGA dans notre cas), constituant ainsi un module matériel qui peut être introduit à n'importe quelle puce pour détecter et localiser les pics thermiques qui peuvent apparaître durant le fonctionnement. Une extension à cet algorithme sera ajoutée afin de le rendre capable de prendre de bonnes décisions de correction et de gestion thermique dynamique.

Dans la première discussion nous allons faire une revue de la méthode GDS (*Gradient Direction Sensor*) qui constitue la base de notre algorithme proposé.

La méthode a été introduite dans [18]. Pour utiliser cette méthode, 4 hypothèses doivent être établies:

1. Il y a seulement une source de chaleur ponctuelle dominante sur la plaque contrôlée.
2. La température est linéairement distribuée par-dessus la surface de la couche de Silicium. Il faut noter que dans ce travail nous considérerons cette supposition.
3. Les détecteurs de température utilisés dans cette méthode donnent des tensions linéairement proportionnelles à la température.
4. Les capteurs dans les 2 cellules des détecteurs sont suffisamment placés proches l'un à l'autre afin que les isothermes les traversant peuvent être représentés par des lignes droites.

Les figures suivantes 2-5 et 2-6 montrent, d'une part la répartition classique des capteurs et d'autre part la répartition proposée des 2 cellules des détecteurs.

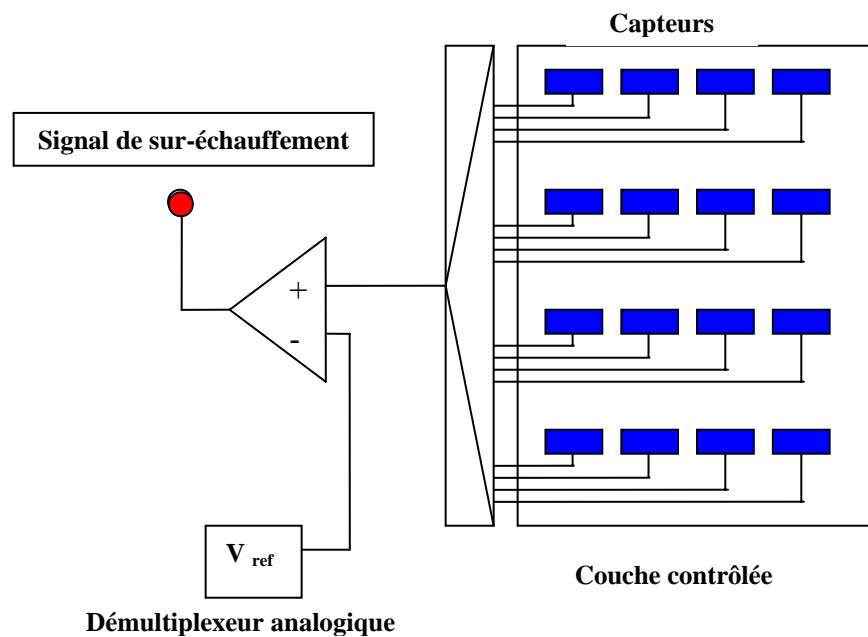


Figure 2.5: Répartition classique

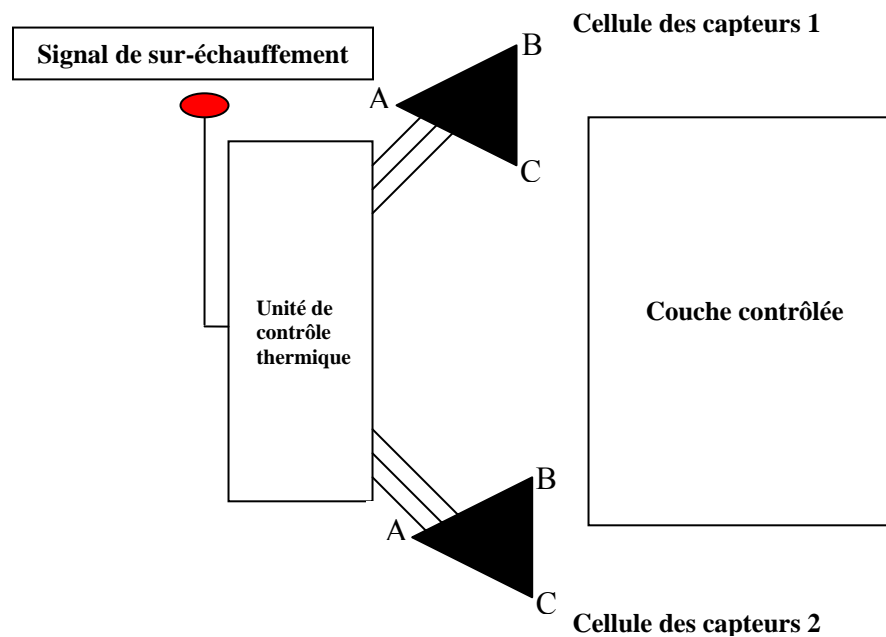
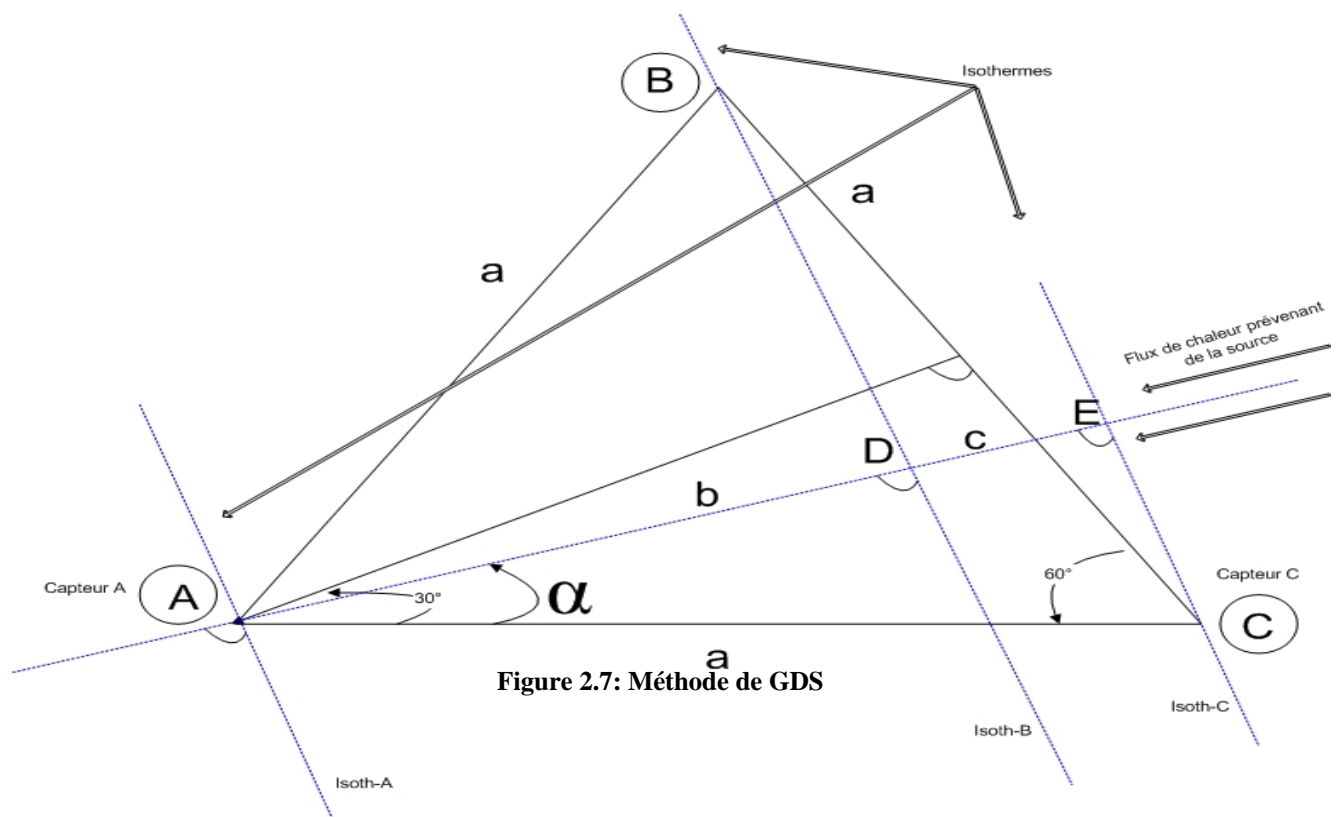


Figure 2.6: Répartition suggérée avec l'unité de contrôle.

On voit bien que la répartition des capteurs minimise la complexité sur la puce. Ces capteurs n'occupent aucun bloc logique sur le FPGA, mais sont placés à l'extérieur de la couche contrôlée.

2.4.1 Description de la méthode GDS

Nous allons maintenant étudier le phénomène physique du GDS qui va nous générer les formules nécessaires pour l'algorithme.



Pour les 2 capteurs A et C placés à une distance a (Fig. 2.7), la différence entre leurs voltages fournies est proportionnelle aux changements de la température de valeur ΔT tout au long de la distance a . Ceci est vrai si et seulement si la source de chaleur est directement sur la ligne AC. Dans tous les autres cas, la valeur de l'angle α entre en action et doit être prise en considération pour calculer ΔT (équation 2.1).

$$\frac{\Delta T}{\Delta r} = \frac{T_C - T_A}{a \cdot \cos \alpha} \leftarrow \text{correspond à} \rightarrow \frac{V_C - V_A}{a \cdot \cos \alpha} \quad (2.1)$$

Où r est la distance de la source de chaleur, V_C et V_A sont les voltages fournis par les capteurs C et A.

Pour obtenir l'angle α on doit entrer en jeu le 3^{ème} capteur qui est B. Dans le cas le plus simple notre méthode de GDS contient seulement 3 capteurs de températures repartis suivant une distance a . Les 3 capteurs ensemble forment la première cellule.

En appliquant les points 2 et 4 de notre hypothèse déjà établit, on peut obtenir les équations suivantes:

$$b = \overline{AD}, \overline{AD} \xleftrightarrow{\text{correspond à}} T_B - T_A \xleftrightarrow{\text{correspond à}} V_B - V_A \quad (2.2)$$

$$b + c = \overline{AE}, \overline{AE} \xleftrightarrow{\text{correspond à}} T_C - T_A \xleftrightarrow{\text{correspond à}} V_C - V_A \quad (2.3)$$

En se basant sur les équations (2.2) et (2.3) et les propriétés géométriques de la figure 2-7 on peut obtenir l'équation (2.4).

$$\tan \alpha \xleftrightarrow{\text{correspond à}} \frac{2}{\sqrt{3}} \cdot \left(\frac{V_B - V_A}{V_C - V_A} - \frac{1}{2} \right) \quad (2.4)$$

Donc en utilisant la cellule des capteurs on peut obtenir de l'information sur la distribution de la température et partiellement sur la position de la source de chaleur.

Pour obtenir la valeur de la température d'une source de chaleur ponctuelle nous devons calculer la distance entre les capteurs et la source de chaleur. Pour cela nous devons introduire une 2^{ème} cellule de capteurs. Les 2 cellules sont placées à une distance donnée H et chacune de ces 2 cellules va nous donner des informations à propos de l'angle α (α_1 et α_2) dans la direction de la source de chaleur (fig.2-8).

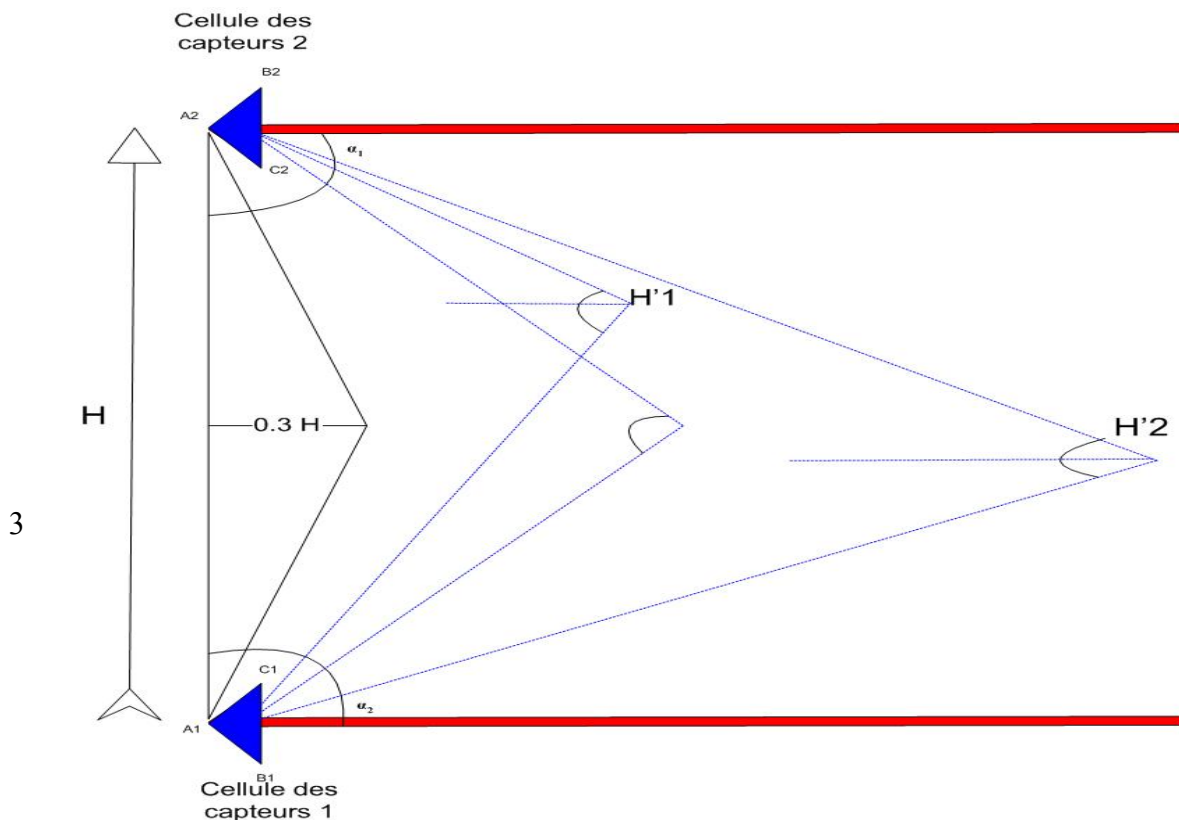


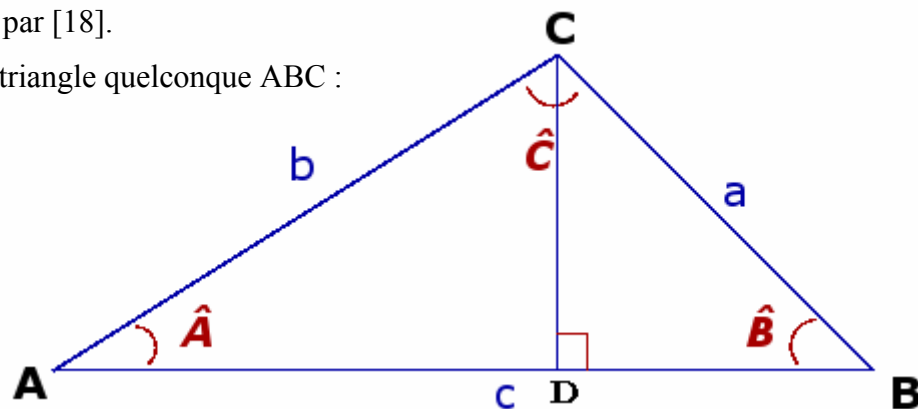
Figure 2.8: Répartition des 2 cellules de capteurs

La source de chaleur et les cellules forment un triangle dans lequel la longueur d'un côté et les valeurs des angles adjacents à ce côté sont connues. Donc, nous pouvons maintenant calculer la distance entre la source de chaleur et les capteurs.

2.4.2 Détermination des équations de la position

Nous pouvons utiliser les propriétés géométriques du triangle formé par les cellules et le centre de la source de la chaleur pour estimer les équations des distances R_1 et R_2 qui n'ont pas été présentées par [18].

Considérons un triangle quelconque ABC :



Supposons que A est le centre du triangle équilatéral formant la première cellule 1, et B le centre de la cellule 2. Dans ce cas C sera le centre de la source S. Selon la géométrie de la figure 2-8, $\tan\alpha_1$ et $\tan\alpha_2$ peuvent être déterminé par l'équation (2.4) et par la suite on peut déterminer α_1 et α_2 . Les angles BAC et ABC sont les compléments des angles α_1 et α_2 . Donc dans le triangle ABC on connaît $AB = H$ et les 3 angles BAC, ABC, ACB alors on peut déterminer les cotés AC (R_1) et BE (R_2).

La loi des sinus s'écrit :

$$\frac{\sin \hat{A}}{a} = \frac{\sin \hat{B}}{b} = \frac{\sin \hat{C}}{c}$$

Donc on peut déduire que : $R_1 = b = \frac{H \sin \hat{B}}{\sin \hat{C}}$ et $R_2 = a = \frac{H \sin \hat{A}}{\sin \hat{C}}$ (2.5)

R_1 et R_2 ont la même unité que H.

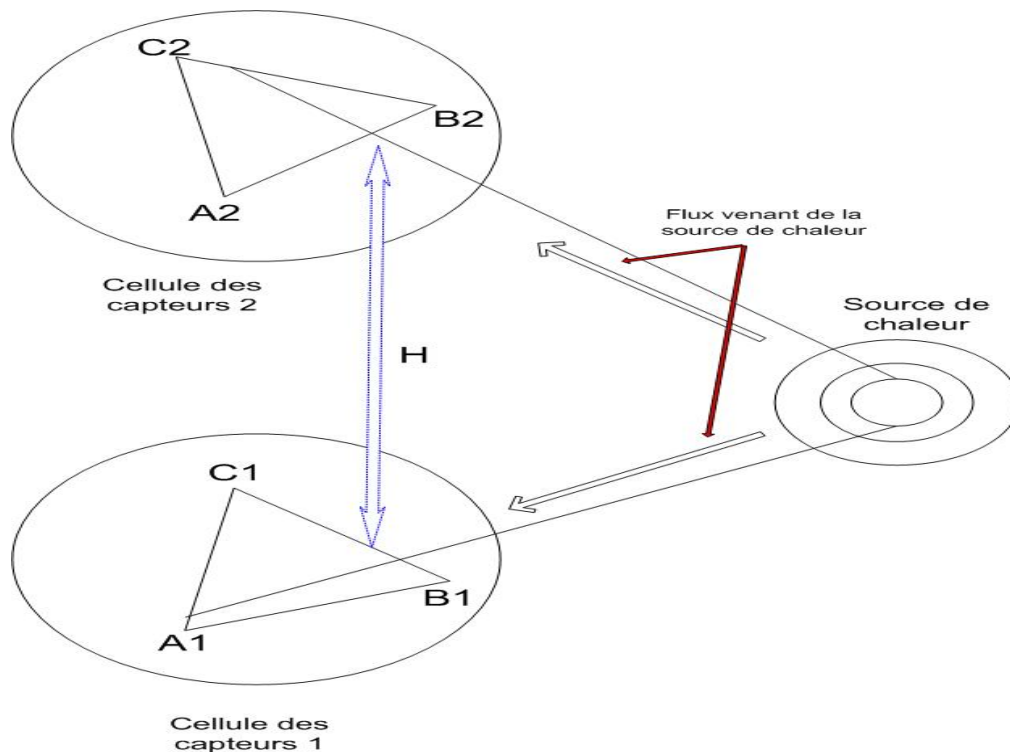


Figure 2.9: Schéma simplifié de la répartition des 2 cellules

La figure 2-9 ci-dessus montre la disposition proposée pour les 2 cellules des capteurs, la cellule 1 comprend les 3 capteurs A1, B1 et C1; la cellule 2 comprend A2, B2 et C2.

Les 2 cellules sont placées sur les 2 coins de la carte à contrôler, et sont séparées par une distance donnée H. On montre ainsi les différentes positions possibles de la source de chaleur sous expérience.

Sur la base des propriétés géométriques de la figure 2-8 et les équations (2.1) (2.2) (2.3) et (2.4) appliquées pour $\alpha \in (-30^\circ, +30^\circ)$ on peut déduire l'équation (2.6) qui calcule la température T_S de la source de chaleur et par la suite son voltage V_S .

$$T_S \xleftrightarrow{\text{Correspond à}} V_S = \frac{H}{a} \cdot (V_{C1} - V_{A1}) \cdot \frac{(\tan^2 \alpha_1 + 1) \cdot (\sqrt{3} + \tan \alpha_2)}{\sqrt{3} \cdot (1 - \tan \alpha_1 \cdot \tan \alpha_2) - (\tan \alpha_1 + \tan \alpha_2)} + V_{A1} \quad (2.6)$$

Cette méthode est très utile pour l'indication des situations de sur-échauffement sur la surface de couches semi-conductrices.

Nous allons implémenter cette méthode directement sur la surface à contrôlée, à cause de 2 grandes avantages représentées par cette méthode et qui sont:

1. L'utilisation de 6 capteurs seulement (dans le cas le plus simple) structurées sous forme de 2 cellules séparées d'une distance H.
2. La distribution de ces 2 cellules peut être à n'importe quel endroit à l'extérieure de la zone étudiée.

L'un des objectifs importants à atteindre est le placement idéal de ces 2 cellules en se basant sur une série de simulations logicielles et matérielles.

L'inconvénient majeur dans cette méthode est principalement dans la complexité du calcul présenté par l'équation (2.6). C'est pour cela que nous proposons l'implantation sur un FPGA, qui se caractérise par des capacités de calcul très élevées.

2.5 Le problème inverse

Le problème inverse en général est la réciproque de la modélisation. Les problèmes Inverses «PI» pourraient être décrits comme des problèmes où on connaît la réponse, mais pas la question, et où les résultats et les conséquences sont déjà déterminés, mais pas la cause. Il existe plusieurs phénomènes physiques qui ne peuvent pas être mesurés directement où ils se manifestent. Nous savons les résultats d'investigation de certains phénomènes, comme par exemple les valeurs de la température dans certaines places sur la surface et nous allons essayer de trouver les paramètres du phénomène source comme la température de la source de chaleur [29] [30].

Par exemple, si 3 ruisseaux se joignent pour former une rivière, et nous savons que 3 usines mettent des quantités connues de polluant dans ces ruisseaux, alors nous pouvons calculer le polluant résultant dans la rivière. Ceci serait l'avant, ou le problème classique. Mais un problème plus probable est que nous savons seulement le niveau total du polluant dans la rivière, et nous devons établir la quantité du polluant de chaque usine dans le ruisseau. Ceci est le problème inverse. Pour mieux comprendre l'explication du problème inverse dans ce qui va suivre nous présentons quelques définitions importantes:

- Paramètres: ils sont le lien entre les données et la variable recherchée.
- Problème mal posé (*ill-posed*): c'est un problème en partie sous-déterminé et en partie sur-déterminé.
- Problème mal conditionné: de petites erreurs provoquent de grandes variations dans le modèle.
- Régularisation: lorsqu'une matrice n'est pas inversible. Il est possible de la rendre inversible en ajoutant une matrice diagonale.
- Convexe: se dit d'une fonction objective qui ne possède qu'un extremum.

L'estimation de flux de chaleur, de positions des sources ou de propriétés thermo physiques de matériaux repose généralement sur la résolution du problème inverse du transfert de la chaleur. De nombreuses applications nécessitent une estimation en temps réel des fonctions

inconnues. La résolution du problème inverse est donc séquentielle et les temps de calculs à chaque itération doivent être petits devant la période d'échantillonnage des mesures. Cela nécessite de disposer d'un modèle simplifié du comportement thermique transitoire du système.

En général, la solution du problème inverse nécessite des efforts de calcul considérables et ne donne pas toujours le résultat exact, ceci est à cause de la nature du problème inverse mal conditionné. Dans certains cas nous pourrions déterminer la fonction de distribution d'un certain phénomène sur la partie du système sous étude. Alors pour obtenir les paramètres de la source nous avons besoin de déterminer à partir d'où le phénomène est distribué et comment il change avec le temps. Les coordonnées géométriques de la source étudiée sont obtenues par l'application de la méthode du GDS [18].

2.6 Conclusion

L'état de l'art nous a montré l'existence de plusieurs bonnes méthodes pour le contrôle et gestion thermique aux niveaux des couches semi-conductrices. Nous avons mis en évidence les avantages et les inconvénients de chacune de ces méthodes. Nous avons montré les fondations de la méthode de GDS choisie, les conditions et contraintes de son application, et introduit le concept du problème inverse qui est à la base de cette méthode

AVIS DE CONFIDENTIALITÉ

À la demande expresse du directeur de recherche et avec le consentement explicite de l'étudiant, la suite de ce mémoire est protégée par une clause de confidentialité.

Extrait de la demande du directeur de recherche cosignée par l'étudiant :

« En raison de la confidentialité du projet de recherche qui s'inscrit dans une perspective de développement d'une unité de contrôle de la contrainte thermomécanique dans les microsystemes VLSI, [il faut] attribuer un caractère confidentiel au contenu du rapport au moins pendant 2 ans après la date à laquelle la Commission des études va recommander le diplôme. »

Bibliographie

- [1] ARRIBAS P. C, HUELIN MACIÁ F. M. *FPGA Implementation of Camus Correlation Optical flow algorithm for Real Time Images*. Dpto. de Sistemas Electrónicos y de Control. E.U.I.T. Telecomunicación et Dpto de Tecnologías Aplicadas a la Telecomunicación. E.T.S.I Telecomunicación. Univ Politécnica de Madrid. 1999.
- [2] BOUGATAYA M, LAKHSASI A, MASSICOTTE D. *VLSI Thermo-mechanical Stress Analysis by Gradient Direction Sensor Method*. IEEE 2005. 0-7803-8886-0/05.
- [3] CHAOUI HICHAM. *Implantation sur FPGA d'une loi de commande adaptative neuronale supervisée pour une articulation flexible*. Projet de Mémoire, Maitrise en Informatique, Université de Québec en Outaouais, 2004. p. 91-108.
- [4] CHAUBAL ADITYA PRAKASH. *Design and Implementation of an FPGA-based Partially Reconfigurable Network Controller*. Master of Science Thesis, Virginia Polytechnic Institute and State University, 2004. p 49-65.
- [5] DRAPER B.A, BEVERIDGE J. R, BÖHM A.P.W, ROSS C. *Accelerated Image Processing on FPGAs work funded by DARPA through AFRL*. 1998.
- [6] JANICKI M, ZUBERT M, NAPIERALSKI A. *Application of inverse problem algorithms for integrated circuit temperature estimation*, Department of Microelectronics and Computer Science, Technical University of Lodz, Al. Politechniki 11, 93-590 Lodz, Poland. 1999 Elsevier Science Ltd. 0026-2692/99.

- [7] KLOTCHKOV I.V, PEDERSEN S. A Codesign Case Study: Implementing Arithmetic Functions in FPGA, Department of Information Technology, Technical University of Denmark DK-2800 Lyngby, Denmark. 1996 IEEE. 0-8186-7355-9/96.
- [8] LOPEZ-BUEDO SERGIO, BOEMO EDUARDO. Making Visible the Thermal Behaviour of Embedded Microprocessors on FPGAs. A Progress Report. Computer Engineering Department, School of Engineering Universidad Autonoma de Madrid, Spain. 2004 ACM 1-58113-829-6/04
- [9] LU WEIDONG. *Designing TCP/IP Functions In FPGAs*. MSc Thesis, Delft University of Technology, Faculty of Electrical Engineering, Mathematics and Computer Science, 2003. p.17.
- [10] MONMASSON E. ET CHAPUIS Y.A. Contributions of FPGAs to the Control of Electrical Systems, a Review. Université de Cergy-Pontoise. 2002.
- [11] MOON Y.S, TANG T.Y, CHAN K.C. Efficient implementation of Fingerprint Verification for Mobile embedded Systems using Fixed-point Arithmetic. ACM Symposium on Applied Computing 2004.
- [12] QUINTON P. ET ROBERT Y. *Algorithmes et architectures systoliques*. Masson 1989. p-6-12.
- [13] RATHA N.K, ROVER D.T. FPGA-based High Performance Page Layout Segmentation. Department of Computer Science and Department of Computer Engineering, Michigan State University, East Lansing, MI48824. IEEE 1996. 0-8186-7502-0/96.
- [14] SENTIEYS OLIVIER, HERVÉ NICOLAS, MÉNARD DANIEL. Synthèse d'architecture sur FPGA sous contrainte de précision des calculs. IRISA – Université de Rennes I.

- [15] SUNDARARAJAN P, MCMILLAN S AND GUCCIONE S.A. *Testing FPGA Devices Using JBits*. Xilinx Inc. 2100 Logic Drive. San Jose, CA 95124 (USA).
- [16] VELUSAMY SIVA, HUANG WEI, LACH JOHN, SKADRON KEVIN. Monitoring Temperature in FPGA Based Socs. Departments of Computer Science, and Electrical and Computer Engineering University of Verginia. CS Technical Report CS-2004-39. IEEE 2005, 0-7695-2451-6/05.
- [17] VICARD A. ET SOREL Y. Optimisation statique d'implantations parallèles. Deuxièmes journées francophones de recherche opérationnelle. Sousse, Tunisie 6-8 Avril 1998.
- [18] WÓJCIAK W AND NAPERALSKI A. *Thermal monitoring of a single heat source in semiconductor devices- the first approach*. Microelectronics Journal 28 (1997) 313-316. Elsevier Science. S0026-2692(96)00034-1.
- [19] YVES SOREL, GRANDPIERRE T., ET LAVARENNE C. Optimized Rapid Prototyping for Real-Time Embedded Heterogeneous Multiprocessors. CODES'99 7th International Workshop on Hardware/Software Co-Design, Rome, May 1999.
- [20] <http://support.xilinx.com/support/support.htm>.
Dernière visite le 6 Juin, 2005.
- [21] <http://homepages.cae.wisc.edu/~ece554/website/>ToolDoc/ToolDoc.html>
Dernière visite le 17 Out, 2005.
- [22] <http://www.ecs.umass.edu/ece/tessier/courses/697ff/lect1-ece697f.pdf>
Dernière visite le 22 Septembre, 2005.
- [23] <http://www.ecs.umass.edu/ece/tessier/courses/697ff/lect10-ece697f.pdf>
Dernière visite le 29 Septembre, 2005.
- [24] <http://www.seas.upenn.edu/~cse371/handouts/ee200.FPGA.pdf>
Dernière visite le 14 Octobre, 2005.

[25] <http://www.cs.colostate.edu/~svr/classes/2001/CS670>

Dernière visite le 18 Octobre, 2005.

[26] <http://www.lri.fr/~de/Reconf.pdf>

Dernière visite le 12 Novembre, 2005.

[27] <http://www.inst.eecs.berkeley.edu/~cs150>

Dernière visite le 28 Décembre, 2005.

[28] <http://www.er.cs.ucla.edu/C2See/docum/c2see.pdf>

Dernière visite le 06 Janvier, 2006.

[29] <http://www.trefle.u-bordeaux1.fr>

Dernière visite le 10 Janvier, 2006.

[30] <http://www.inverse-problems.com>

Dernière visite le 10 Janvier, 2006.

[31] <http://srdata.nist.gov/its90/download/allcoeff.tab>

Dernière visite le 15 Juin 2006.

[32] <http://www.a525g.com/mathematiques/moindres-carres.php>

Dernière visite le 20 Juin 2006.

[33] <http://lmi17.cnam.fr/~anceau/Documents/ECSCONS.pdf>

Dernière visite le 21 juin 2006.

[34] <http://www.essi.fr>

Dernière visite le 21 juin 2006.